Towards Automatic Debugging of Computer Programs (1991) (Make Corrections) (17 citations) Hiralal Agrawal

CiteSeer Home/Search Bookmark Context Related

View or download:

purdue.edu/pub/serc/tech...TR103P.PS.Z purdue.edu/pub/serc/tech...TR103P.PS.Z purdue.edu/pub/spaf...agrawal1991.ps.Z Cached: PS.gz PS PDF Image Update Help

From: <u>purdue.edu (more)</u>
From: <u>rzaix340.rz.unileipzig.</u> <u>slicing</u>
(<u>Enter author homepages</u>)

(Enter summary)

Rate this article: 1 2 3 4 5 (best)

Comment on this article

Abstract: Programmers spend considerable time debugging code. Symbolic debuggers provide some help but the task still remains complex and difficult. Other than breakpoints and tracing, these tools provide little high level help. Programmers must perform many tasks manually that the tools could perform automatically, such as finding which statements in the program affect the value of an output variable under a given testcase, what was the value of a given variable when the control last reached a given... (Update)

Context of citations to this paper: More

...the computation is almost correct such information will be of little help in determining where the error occurs. **Program slicing [Agr91, Pan93, Wei82] is a way of determining which lines of code affect a certain variable.** Thus if one knows that a certain variable is...

.... occurrences of a statement is represented by more than one vertex only if the two executions have different transitive dependences [Agr91]. The PDG can be augmented with other kinds of edges to represent other dependences. To represent a program with procedures we need...

Cited by: More

Debugging with Dynamic Slicing and Backtracking - Agrawal, Demillo, Spafford (1993) (Correct)

A Hybrid Program Slicing Framework - Rilling, Karanth (2001) (Correct)

Points-to Analysis for Program Understanding - Fiutem, Tonella, Antoniol, Merlo (1997) (Correct)

Similar documents (at the sentence level):

6.4%: Dynamic Slicing in the Presence of Unconstrained Pointers - Agrawal (1991) (Correct)

Active bibliography (related documents): More All

- 2.8: Software Debugging With Dynamic Instrumentation And Test-Based.. Pan (1993) (Correct)
- 2.3: Fault Investigation and Trial Viravan (1991) (Correct)
- 2.1: Enhancing Debugging Technology Viravan (1994) (Correct)

Similar documents based on text: More All

- 0.3: An Execution Backtracking Approach to Program Debugging Agrawal, DeMillo, Spafford (1991) (Correct)
- 0.3: Compiler-Integrated Software Testing Krauser, Jr. (1991) (Correct)
- 0.2: X-Gen: A Random Test-Case Generator For Systems And Socs Emek, Jaeger, Naveh.. (2002) (Correct)

Related documents from co-citation: More All

- 10: Programmers use slices when debugging (context) Weiser 1982
- 10: IEEE Transactions on Software Engineering (context) Weiser 1984
- 10: An execution backtracking approach to program debugging Agrawal, DeMillo et al. 1991

BibTeX entry: (Update)

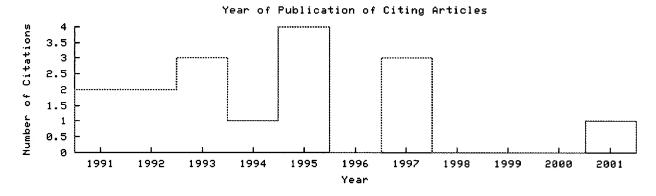
Hiralal Agrawal. Towards Automatic Debugging of Computer Programs. PhD thesis, Department of Computer Sciences, Purdue University, West Lafayette, IN, 1991. http://citeseer.ist.psu.edu/agrawal91towards.html More

```
@techreport{ agrawal91towards,
    author = "Hiralal Agrawal",
    title = "Towards Automatic Debugging of Computer Programs",
    number = "Ph.D. Thesis",
    year = "1991",
    url = "citeseer.ist.psu.edu/agrawal91towards.html" }
```

Citations (may not include all citations):

- 1020 The Definition of Standard ML (context) Milner, Tofte et al. 1990
- 621 A Discipline of Programming (context) Dijkstra 1976
- 617 An axiomatic basis for computer programming (context) Hoare 1969
- 447 The program dependence graph and its uses in optimization (context) Ferrante, Ottenstein et al. 1987
- 417 Software Engineering Economics (context) Boehm 1981
- 354 Interprocedural slicing using dependence graphs Horwitz, Reps et al. 1990
- 320 Algorithmic Program Debugging (context) Shapiro 1983
- 302 IEEE Transactions on Software Engineering (context) Weiser 1984
- 290 Free Software Foundation (context) Stallman, Porting et al. 1990
- 290 Free Software Foundation (context) Stallman, third et al. 1989
- 227 Analysis of pointers and structures (context) Chase, Wegman et al. 1990
- 172 The program dependence graph in a software development envir.. (context) Ottenstein, Ottenstein 1984
- 168 Dependence graphs and compiler optimizations (context) Kuck, Kuhn et al. 1981
- 127 Detecting conflicts between structure accesses (context) Larus, Hilfinger 1988
- 122 Debugging concurrent programs (context) McDowell, Helmbold 1989
- 115 Programmers use slices when debugging (context) Weiser 1982
- 97 Algorithm Animation (context) Brown 1988
- 94 Dependence analysis for pointer variables (context) Horwitz, Pfeiffer et al. 1989
- 77 A formal model of program dependences and its implications f.. (context) Podgurski, Clarke 1990
- 72 The semantic approach to program slicing (context) Venkatesh 1991
- 70 ACM Transactions on Programming Languages and Systems (context) Hennessy, of 1982
- 64 The Cornell Program Synthesizer: a syntax-directed programmi . (context) Teitelbaum, Reps 1981
- 62 Techniques for algorithm animation (context) Brown, Sedgewick 1985
- 62 Information Processing Letters (context) Korel, Laski et al. 1988
- 53 Automatic program bug location by program slicing (context) Lyle, Weiser 1987
- 53 IEEE standard glossary of software engineering terminology (context) IEEE 1983
- 48 A mechanism for efficient debugging of parallel programs (context) Miller, Choi 1988
- 46 Theory of Recursive Functions and Effective Computability (context) Rogers 1967
- 45 Provide: a process visualization and debugging environment (context) Moher 1988
- 38 The relationship between slices and module cohesion Ott, Thuss 1989
- 38 The Programmer's Apprentice: A research overview (context) Rich, Waters 1988
- 37 Dynamic slicing of computer programs (context) Korel, Laski 1990
- 37 The semantics of program dependence Cartwright, Felleisen 1989
- 34 Doc; a practical approach to source-level debugging of globa.. (context) Coutant 1988
- 32 Social processes and the proofs of theorems and programs (context) DeMillo, Lipton et al. 1979
- 31 Animating programs using Smalltelk (context) London, Duisberg 1985
- 31 User recovery and reversal in interactive systems Archer, Richard et al. 1984
- 26 ACM Transactions on Programming Languages and Systems (context) Horwitz, Prins et al. 1989
- 26 Data flow analysis in software reliability (context) Fosdick, Osterweil 1976
- 23 Proust: Knowledge-based program understanding (context) Johnson, Soloway 1985
- 22 Functional Program Testing and Analysis (context) Howden 1987
- 21 Program verification; the very idea (context) Fetzer 1988
- 19 the use of software artifacts to evaluate the effectiveness .. (context) DeMillo, Mathur 1991
- 18 Linked-list visualization for debugging (context) Shimomura, Isoda 1991
- 17 Obxtoot: a window-based symbolic debugger for Sun work-stati.. (context) Adams, Muchnick 1986
- 16 The relationship between program complexity and slice complet. (context) Longworth, Ott et al. 1986
- 15 Vips: a visual debugger (context) Isoda, Shimomura et al. 1987
- 15 Xerox Palo Alto Research Center (context) Teitelman, Manual et al. 1978
- 14 Knowledge-based program debugging systems (context) Seviora 1987
- 14 Software aspects of strategic defense systems (context) Parnas 1985
- 12 A rewriting semantics for program dependence graphs (context) Selke 1989
- 11 Vax DEBUG: an interactive (context) Beander 1983
- 11 Debugging with Dbx (context) Dunlap 1986
- 11 Stad---a system for testing and debugging: User perspective (context) Korel, Laski 1988
- 10 Understanding and debugging programs (context) Lukey 1980
- 9 Two-dimensional pinpointing: Debugging with formal specifica.. (context) Luckham, Sankar et al. 1991
- 8 Some psychological evidence on how people debug computer pro.. (context) Gould 1975
- 8 A futorial introduction to ADS (context) Maranzano, Bourne 1979
- 7 Exdams, Extendible debugging and monitoring system (context) Balzer 1969
- 7 Compiler-integrated program mutation (context) DeMillo, Krauser et al. 1991
- 7 Perspectives on software engineering (context) Zelkowitz 1978
- 7 An overview of bugs (context) Schwartz 1971
- 7 Towards automatic software fault location through decision t.. (context) Collofello, Cousins 1987
- 6 Computer Science Department (context) Reps, Yang et al. 1988
- 5 Knowledge-based fault localization in debugging (context) Sedlmeyer, Thompson et al. 1983

- 5 Flit-flexowriter interrogation tape: a symbolic utility fo., (context) Stockham, Dennis 1960
- 5 Information-flow and data-flow analysis of while programs (context) Bergeretti, Carr'e 1985
- 5 A program decomposition scheme with applications to software.. (context) Gallagher, Lyle 1989
- 5 DAVE—a validation error detection and documentation system. (context) Osterweil, Fosdick 1976
- 5 Detection of data flow anomaly through program instrumentati.. (context) Huang 1979
- 4 ACM Software Engineering Notes (context) Agrawal, Spafford et al. 1989
- 4 Reversible Execution As a Diagnostic Tool (context) Zelkowitz 1971
- 4 line debugging systems (context) Saunders, Wagner 1965
- 4 Sequential debugging at a high level of abstraction (context) Olsson, Crawford et al. 1991
- 3 AIDA---a dynamic data flow anomaly detection system for Pasc., (context) Chan, Chen 1987
- 3 The feel of Pi (context) Cargill 1986
- 3 Algorithmic software fault localization (context) Korel, Laski 1991
- 3 Semi-automatic bug localization in software maintenance (context) Shahmehri, Kamkar et al. 1990
- 3 A review of automated debugging systems; Knowledge (context) Ducasse, Emde 1988
- 2 Compiler-Integrated Software testing Krauser 1991
- 2 Unix Programmer's Manual (context) Katsoff, debugger 1979
- 1 Journal of Systems and Software (context) Cargill, debugger 1983
- 1 Scientific Affairs Div (context) Nauer, Randell et al. 1969
- 1 on and M. A. Branstad, Validation, verification, and testing., (context) Adri 1982
- 1 Cummings Publishing Co (context) DeMillo, McCracken et al. 1987
- 1 Automated programming: The Programmer's Assistant (context) Teitelman 1972



The graph only includes citing articles where the year of publication is known.

Documents on the same site (http://fermivista.math.jussieu.fr/ftp/ftp.cs.purdue.edu.html): More A Formal Evaluation of Mutation and Data Flow Based Test.. - Mathur, Wong (1994) (Correct)
Smooth Blending of Basic Surfaces using Trivariate Box Splines - Peters, Wittman (1996) (Correct)
Pattern Matching Image Compression: Algorithmic and.. - Atallah, Genin.. (1995) (Correct)

CiteSeer.IST - Copyright Penn State and NEC

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library

automatic bug location <and> program slicing



Feedback Report a problem Satisfaction SUIVEV

Terms used automatic bug location and program slicing

Found 157,162 of 851,460

Sort results by Display

results

relevance expanded form

Save results to a Binder Search Tips Copen results in a new

Try an Advanced Search Try this search in The Digital Library

Results 1 - 20 of 200

window

Result page: 1 2 3 4 5 6 7 8 9 10

Relevance scale 🔲 📟 📟

Best 200 shown

Generalized algorithmic debugging and testing

Peter Fritzson, Nahid Shahmehri, Mariam Kamkar, Tibor Gyimothy

December 1992 ACM Letters on Programming Languages and Systems (LOPLAS), Volume 1 Issue 4

Full text available: pdf(1.35 MB)

Additional Information: foli citation, abstract, references, citings, index

terms

This paper presents a method for semi-automatic bug localization, generalized algorithmic debugging, which has been integrated with the category partition method for functional testing. In this way the efficiency of the algorithmic debugging method for bug localization can be improved by using test specifications and test results. The long-range goal of this work is a semi-automatic debugging and testing system which can be used during largescale program development of nontrivial programs. ...

Keywords: algorithmic debugging, automated debugging, category partition testing, program slicing

Tracking pointers with path and context sensitivity for bug detection in C programs V. Benjamin Livshits, Monica S. Lam



September 2003 ACM SIGSOFT Software Engineering Notes, Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering, Volume 28 Issue 5

Full text available: pdf(220.74 KB)

Additional Information: full citation, abstract, references, citings, index terms

This paper proposes a pointer alias analysis for automatic error detection. State-of-the-art pointer alias analyses are either too slow or too imprecise for finding errors in real-life programs. We propose a hybrid pointer analysis that tracks actively manipulated pointers held in local variables and parameters accurately with path and context sensitivity and handles pointers stored in recursive data structures less precisely but efficiently. We make the unsound assumption that pointers passed i ...

Keywords: SSA representation, buffer overruns, context-sensitive analysis, error detection, path-sensitive analysis, pointer analysis, program analysis, program representation, security flaws, software security

Generalized algorithmic debugging and testing Peter Fritzson, Tibor Gyimothy, Mariam Kamkar, Nahid Shahmehri May 1991 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLAN 1991 conference on Programming language design and implementation. Volume 26 Issue 6



Full text available: pdf(1 00 MB)

Additional Information: full citation, references, citings, index terms

A review of automated debugging systems: knowledge, strategies and techniques M. Ducassé, A.-M. Emde



April 1988 Proceedings of the 10th international conference on Software engineering

Full text available: pdf(982.37 KB)

Additional Information: full citation, abstract, references, citings, index

Our review is based on descriptions of 18 existing automated systems on program debugging and of a dozen cognitive studies on debugging. We propose a classification of debugging knowledge, and a description of the corresponding knowledge representation in the systems. Then we propose a classification of global debugging strategies used in the systems, and a description of the corresponding techniques. We assess the identified strategies from a real world program development point of view.

5 Dynamic program slicing

Hiralal Agrawal, Joseph R. Horgan

June 1990 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLAN 1990 conference on Programming language design and implementation, Volume 25 Issue 6

Full text available: Todf(1.11 MB)

Additional Information: foll citation, abstract, references, citings, index terms

Program slices are useful in debugging, testing, maintenance, and understanding of programs. The conventional notion of a program slice, the static slice, is the set of all statements that might affect the value of a given variable occurrence. In this paper, we investigate the concept of the dynamic slice consisting of all statements that actually affect the value of a variable occurrence for a given program input. The sens ...

A Fast Assembly Level Reverse Execution Method via Dynamic Slicing May 2004 Proceedings of the 26th International Conference on Software Engineering





Additional Information: full citation, abstract

One of the most time consuming parts of debugging istrying to locate a bug. In this context, there are two powerfuldebugging aids which shorten debug time considerably:reverse execution and dynamic slicing. Reverse executioneliminates the need for repetitive program restartsevery time a bug location is missed. Dynamic slicing, onthe other hand, isolates code parts that influence an erroneous variable at a program point. In this paper, we presentan approach which provides assembly level reverse ex ...

Static and Dynamic Slicing of Constraint Logic Programs Gyöngyi Szilágyi, Tibor Gyimóthy, Jan Małuszyński

January 2002 Automated Software Engineering, Volume 9 Issue 1



Additional Information: full citation, abstract, index terms

Slicing is a program analysis technique originally developed for imperative languages. It facilitates understanding of data flow and debugging.

This paper discusses slicing of Constraint Logic Programs. Constraint Logic Programming (CLP) is an emerging software technology with a growing number of applications. Data flow in constraint programs is not explicit, and for this reason the concepts of slice and the slicing techniques of imperative languages are not directly applicable.

T ...

Keywords: Constraint Logic Programming (CLP), Logic Programming (LP), debugging, program analysis, slicing

Dynamic program analysis: Isolating cause-effect chains from computer programs Andreas Zeller



November 2002 Proceedings of the 10th ACM SIGSOFT symposium on Foundations of software engineering

Full text available: pdf(613.08 KB)

Additional Information: full citation, abstract, references, citings, index terms

Consider the execution of a failing program as a sequence of program states. Each state induces the following state, up to the failure. Which variables and values of a program state are relevant for the failure? We show how the Delta Debugging algorithm isolates the relevant variables and values by systematically narrowing the state difference between a passing run and a failing run---by assessing the outcome of altered executions to determine wether a change in the program state makes a ...

Keywords: automated debugging, program comprehension, testing, tracing

Session 2: dynamic program analysis: Isolating cause-effect chains from computer programs



Andreas Zeller

November 2002 ACM SIGSOFT Software Engineering Notes, Volume 27 Issue 6

Full text available: pdf(1.12 MB)

Additional Information: full citation, abstract, references, index terms

Consider the execution of a failing program as a sequence of program states. Each state induces the following state, up to the failure. Which variables and values of a program state are relevant for the failure? We show how the Delta Debugging algorithm isolates the relevant variables and values by systematically narrowing the state difference between a passing run and a failing run--by assessing the outcome of altered executions to determine wether a change in the program state makes a d ...

Keywords: automated debugging, program comprehension, testing, tracing

10 Program slicing

Mark Weiser

March 1981 Proceedings of the 5th international conference on Software engineering

Full text available: pdf(939.04 KB)

Additional Information: full citation, abstract, references, citings, index terms

Program slicing is a method used by experienced computer programmers for abstracting from programs. Starting from a subset of a program's behavior, slicing reduces that program to a minimal form which still produces that behavior. The reduced program, called a "slice", is an independent program guaranteed to faithfully represent the original program within the domain of the specified subset of behavior. Finding a slice is in general unsolvable. A dataflow algorithm is ...

Keywords: Data flow analysis, Debugging, Human factors, Program maintenance, Program metrics, Software tools

11 Critical slicing for software fault localization

Richard A. DeMillo, Hsin Pan, Eugene H. Spafford

May 1996 ACM SIGSOFT Software Engineering Notes, Proceedings of the 1996 ACM SIGSOFT international symposium on Software testing and analysis, Volume 21 Issue 3

Full text available: pdf(1.33 MB)

Additional Information: full citation, abstract, references, citings, index terms

Developing effective debugging strategies to guarantee the reliability of software is important. By analyzing the debugging process used by experienced programmers, we have found that four distinct tasks are consistently performed: (1) determining statements





involved in program failures, (2) selecting suspicious statements that might contain faults. (3) making hypotheses about suspicious faults (variables and locations), and (4) restoring program state to a specific statement for verification, T ...

Keywords: critical slicing, debugging, dynamic program slicing, failures, fault localization, faults, mutation analysis, static program slicing, testing

12 Techniques for debugging parallel programs with flowback analysis

Jong-Deok Choi, Barton P. Miller, Robert H. B. Netzer

October 1991 ACM Transactions on Programming Languages and Systems (TOPLAS),

Volume 13 Issue 4 Full text available: pdf(2.73 MB)

Additional Information: full citation, references, citings, index terms

Keywords: debugging, flowback analysis, incremental tracing, parallel program, program dependence graph, semantic analysis

13 Slicing spreadsheets: an integrated methodology for spreadsheet testing and debugging



December 1999 ACM SIGPLAN Notices, Proceedings of the 2nd conference on Domainspecific languages, Volume 35 Issue 1

Full text available: pdf(2.17 MB)

Additional Information: full citation, abstract, references, citings, index

Spreadsheet languages, which include commercial spreadsheets and various research systems, have proven to be flexible tools in many domain specific settings. Research shows, however, that spreadsheets often contain faults. We would like to provide at least some of the benefits of formal testing and debugging methodologies to spreadsheet developers. This paper presents an integrated testing and debugging methodology for spreadsheets. To accommodate the modeless and incremental development, t ...

14 Programmers use slices when debugging

Mark Weiser

July 1982 Communications of the ACM, Volume 25 Issue 7

Full text available: pdf(709.05 KB)

Additional Information: full citation, abstract, references, citings, index terms

Computer programmers break apart large programs into smaller coherent pieces. Each of these pieces: functions, subroutines, modules, or abstract datatypes, is usually a contiquous piece of program text. The experiment reported here shows that programmers also routinely break programs into one kind of coherent piece which is not coniquous. When debugging unfamiliar programs programmers use program pieces called slices which are sets of statements related by their flow of dat ...

Keywords: program decomposition, slice

15 Error explanation: PSE: explaining program failures via postmortem static analysis Roman Manevich, Manu Sridharan, Stephen Adams

October 2004 Proceedings of the 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering

Full text available: pdf(218.09 KB) Additional Information: full citation, abstract, references, index terms

In this paper, we describe PSE (Postmortem Symbolic Evaluation), a static analysis algorithm that can be used by programmers to diagnose software failures. The algorithm requires minimal information about a failure, namely its kind (e.g. NULL dereference), and its location in the program's source code. It produces a set of execution traces along which





the program can be driven to the given failure.

PSE tracks the flow of a single value of interest from the point in the program where th ...

Keywords: alias analysis, postmortem analysis, typestate, value flow

16 Experimental Evaluation of Program Slicing for Fault Localization Shinji Kusumoto, Akira Nishimatsu, Keisuke Nishie, Katsuro Inoue



March 2002 Empirical Software Engineering, Volume 7 Issue 1

Full text available: Additional Information: full citation, abstract, index terms

Debugging large and complex software systems requires significant effort since it is very difficult to localize and identify faults. Program slicing has been proposed to efficiently localize faults in the program. Despite the fact that a number of debug systems using program slicing, have been developed, the usefulness of this method to fault localization has not been sufficiently evaluated. This paper aims to experimentally evaluate the usefulness of the program slicing method to fault local ...

Keywords: Program slice, empirical evaluation, fault localization, measurement, tool

17 Syntax-Directed Amorphous Slicing



Mark Harman, Lin Hu, Malcolm Munro, Xingyuan Zhang, Dave Binkley, Sebastian Danicic, Mohammed Daoudi, Lahcen Ouarbya

January 2004 Automated Software Engineering, Volume 11 Issue 1

Full text available: Publisher Sile Additional

Additional Information: full citation, abstract, index terms

An amorphous slice of a program is constructed with respect to a set of variables. The amorphous slice is an executable program which preserves the behaviour of the original on the variables of interest. Unlike syntax-preserving slices, amorphous slices need not preserve a projection of the syntax of a program. This makes the task of amorphous slice construction harder, but it also often makes the result thinner and thereby preferable in applications where syntax preservation is unimportant.<...

Keywords: FermaT, WSL, amorphous slicing, conditioned slicing, transformation

18 Aspect: detecting bugs with abstract dependences



Daniel Jackson

April 1995 ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 4 Issue 2

Full text available: pdf(2.44 MB)

Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index</u> <u>terms</u>

Aspect is a static analysis technique for detecting bugs in imperative programs, consisting of an annotation language and a checking tool. Like a type declaration, an Aspect annotation of a procedure is a kind of declarative, partial specification that can be checked efficiently in a modular fashion. But instead of constraining the types of arguments and results, Aspect specifications assert dependences that should hold between inputs and outputs. The checker uses a simple dependence analys ...

Keywords: dataflow dependences, partial specification, partial verification

19 A mechanism for efficient debugging of parallel programs



June 1988 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLAN 1988 conference on Programming Language design and Implementation, Volume 23 Issue 7

Full text available: Todf(1,29 MB)

Additional Information: full citation, abstract, references, citings, index terms

This paper addresses the design and implementation of an integrated debugging system for parallel programs running on shared memory multi-processors (SMMP). We describe the use of flowback analysis to provide information on causal relationships between events in a program's execution without re-executing the program for debugging. We introduce a mechanism called incremental tracing that, by using semantic analyses of the debugged program, makes the flowback ...

20 A mechanism for efficient debugging of parallel programs



Barton P. Miller, Jong-Deok Choi

November 1988 ACM SIGPLAN Notices, Proceedings of the 1988 ACM SIGPLAN and SIGOPS workshop on Parallel and distributed debugging, Volume 24 Issue 1

Full text available: Pdf(1.22 MB)

Additional Information: full citation, abstract, references, citings, index terms, review

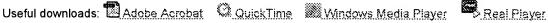
This paper addresses the design and implementation of an integrated debugging system for parallel programs running on shared memory multi-processors (SMMP). We describe the use of flowback analysis to provide information on causal relationships between events in a program's execution without re-executing the program for debugging. We introduce a mechanism called incremental tracing that, by using semantic analyses of the debugged program, makes the flowback ...

Results 1 - 20 of 200

Result page: 1 2 3 4 5 6 7 8 9 10 next

The ACM Portal is published by the Association for Computing Machinery. Copyright @ 2005 ACM, Inc. Terms of Usage Privacy Policy Code of Ethics Contact Us







Debugging with Dynamic Slicing and Backtracking - Agrawal, Demillo, Spafford (ResearchIndex)

Page 1 of 3

Debugging with Dynamic Slicing and Backtracking (1993) (Make

Corrections) (49 citations)

Hiralal Agrawal, Richard A. Demillo, Eugene H. Spafford Software - Practice and Experience

CiteSeer Home/Search Bookmark Context Related

From: cs.ubc.ca/local/reading/...spetoc (more)
(Enter author homepages)

(Enter summary)

Rate this article: 1 2 3 4 5 (best)

Comment on this article

Abstract: this paper we present a debugging model, based on dynamic program slicing and execution backtracking techniques, that easily lends itself to automation. This model is based on experience with using these techniques to debug software. We also present a prototype debugging tool, SPYDER, that explicitly supports the proposed model, and with which we are performing further debugging research (Update)

Context of citations to this paper: More

...slice. Since debugging typically takes place after a program has been executed, dynamic slicing is particularly useful in debugging [1, 36], 2003 John Wiley Sons, Ltd. Softw. Pract. Exper. Conditioned slicing was introduced by Canfora et al. 8] in 1994. Field, Ramalingam...

Cited by: More

Slicing Concurrent Java Programs: Issues and Solutions - Nanda (2001) (Correct)

Efficient Forward Computation of Dynamic Slices Using - Reduced Ordered Binary (Correct)

Cost Effective Dynamic Program Slicing - Xiangyu Zhang Rajiv (2004) (Correct)

Similar documents (at the sentence level):

55.1%: Debugging with Dynamic Slicing and Backtracking - Agrawal, DeMillo, Spafford (1993) (Correct)

19.3%: Towards Automatic Debugging of Computer Programs - Agrawal (1991) (Correct)

7.7%: Efficient Debugging with Slicing and Backtracking - Agrawal, DeMillo, Spafford (1990) (Correct)

Active bibliography (related documents): More All

- 1.4: Software Debugging With Dynamic Instrumentation And Test-Based.. Pan (1993) (Correct)
- 1.1: Enhancing Debugging Technology Viravan (1994) (Correct)
- 1.0: Fault Investigation and Trial Viravan (1991) (Correct)

Similar documents based on text: More All

- 0.7: Dynamic Slicing in the Presence of Unconstrained Pointers Agrawal, DeMillo, Spafford (1991) (Correct)
- 0.7: An Execution Backtracking Approach to Program Debugging Agrawal, DeMillo, Spafford (1991) (Correct)
- 0.6: Heuristics for Automatic Localization of Software Faults Pan, Spafford (1992) (Correct)

Related documents from co-citation: More All

- 33: Interprocedural slicing using dependence graphs Horwitz, Reps et al. 1990
- 27: Using Program Slicing in Software Maintenance Gallagher, Lyle 1991
- 23: The program dependence graph and its use in optimization (context) Ferrante, Ottenstein et al. 1987

BibTeX entry: (Update)

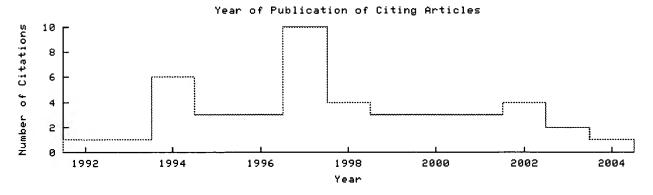
H. Agrawal, R. Demillo, and E. Spafford, "Debugging with Dynamic Slicing and Backtracking," Software-Practice and Experience, Vol.23, No.6, pp.589-616, 1993. http://citeseer.ist.psu.edu/agrawal93debugging.html More

```
@article{ agrawal93debugging,
    author = "Hiralal Agrawal and Richard A. DeMillo and Eugene H. Spafford",
    title = "Debugging with Dynamic Slicing and Backtracking",
    journal = "Software - Practice and Experience",
    volume = "23",
    number = "6",
    pages = "589-616",
    year = "1993",
    url = "citeseer.ist.psu.edu/agrawal93debugging.html" }
```

Citations (may not include all citations):

447 The program dependence graph and its uses in optimization (context) - Ferrante, Ottenstein et al. - 1987

- 354 Interprocedural slicing using dependence graphs Horwitz, Reps et al. 1990
- 290 Free Software Foundation (context) Stallman, Porting et al. 1990
- * 290 Free Software Foundation (context) Stallman, Manual et al. 1989
- 172 The program dependence graph in a software development envir.. (context) Ottenstein, Ottenstein 1984
- 124 Integrating noninterfering versions of programs (context) Horwitz, Prins et al. 1989
- 122 Debugging concurrent programs (context) McDowell, Helmbold 1989
- 120 Dynamic program slicing Agrawal, Horgan 1990
- 115 Programmers use slices when debugging (context) Weiser 1982
- 72 The semantic approach to program slicing (context) Venkatesh 1991
- 64 The Cornell Program Synthesizer: a syntax-directed programmi.. (context) Teitelbaum, Reps 1981
- 53 Automatic program bug location by program slicing (context) Lyle, Weiser 1987
- 48 A mechanism for efficient debugging of parallel programs (context) Miller, Choi 1988
- 40 ligor: a system for program debugging via reversible executio.. (context) Feldman, Brown 1989
- 38 The relationship between slices and module cohesion Ott, Thuss 1989
- 37 The semantics of program dependence Cartwright, Felleisen 1989
- 37 Oynamic slicing of computer programs (context) Korel, Laski 1990
- 35 Dynamic slicing in the presence of unconstrained pointers Agrawal, DeMillo et al. 1991
- 31 User recovery and reversal in interactive systems Archer, Richard et al. 1984
- 23 The semantics of program slicing (context) Reps, Yang 1988
- 20 Using program slicing in software maintenance (context) Gallagher, Lyle 1990
- 18 Demonic memory for process histories (context) Wilson, Moher 1989
- 17 Towards automatic debugging of computer programs Agrawal 1991
- 17 Dbxtooi; a window-based symbolic debugger for Sun workstalio.. (context) Adams, Muchnick 1986
- 17 An execution backtracking approach to program debugging Agrawal, DeMillo et al. 1991
- 16 The relationship between program complexity and slice comple. (context) Longworth, Ott et al. 1986
- 15 Xerox Palo Alto Research Center (context) Teitelman, Manual et al. 1978
- 12 A rewriting semantics for program dependence graphs (context) Selke 1989
- 11 Debugging with Dbx (context) Dunlap 1986
- 10 Understanding and debugging programs (context) Lukey 1980
- 10 A general framework for debugging (context) Araki, Furukawa et al. 1991
- 8 Program slicing Weiser 1984
- 8 A tutorial introduction to ADB (context) Maranzano, Bourne 1979
- 8 Some psychological evidence on how people debug computer pro.. (context) Gould 1975
- 7 An overview of bugs (context) Schwartz 1971
- 7 Compiler-integrated program mutation (context) DeMillo, Krauser et al. 1991
- 7 Exdams: extendible debugging and monitoring system (context) Balzer 1969
- 7 Pi; a case study in object-oriented programming (context) Cargill 1986
- 4 Sequential debugging at a high level of abstraction (context) Olsson, Crawford et al. 1991
- 4 Sdb: a symbolic debugger (context) Katsoff 1979
- 4 Towards automatic localization of software faults (context) Pan, Spafford 1992
- 4 Information-flow and data-flow analysis of while programs (context) Bergeretti, Carre 1985
- 4 Reversible execution as a diagnostic tool (context) Zelkowitz 1971
- 2 A bibliography on debugging and backtracking (context) Agrawal, Spafford 1989
- 2 Compiler-integrated software testing Krauser 1991



The graph only includes citing articles where the year of publication is known.

Documents on the same site (http://www.cs.ubc.ca/local/reading/proceedings/spe91-95/spe/spetoc.htm): <u>More</u> Linkage Analysis of Processes - Yaung, Raz (1992) (<u>Correct)</u>

A Scheme for Little Languages in Interactive Graphics - Beckman (1991) (Correct)

A Dataflow Approach to Event-based Debugging - Olsson, Crawford, Ho (1991) (Correct)

Debugging with Dynamic Slicing and Backtracking - Agrawal, Demillo, Spafford (ResearchIndex)

Page 3 of 3

Online articles have much greater impact More about CiteSeer.IST Add search form to your site Submit documents
*Feedback

CiteSeer.IST - Copyright Penn State and NEC